

LEARNING DESIGN AND SERVICE-ORIENTED ARCHITECTURES: A MUTUAL DEPENDENCY?

Patrick McAndrew

Institute of Educational Technology
The Open University, UK
p.mcandrew@open.ac.uk

Martin Weller

Institute of Educational Technology
The Open University, UK
m.j.weller@open.ac.uk

Mark Barrett-Baxendale

Learning Technologies Research Group
Deanery of Business and Computer Sciences
Liverpool Hope University, UK
BARRETM@hope.ac.uk

Abstract

This paper looks at how the concept of reusability has gained currency in e-learning. Initial attention was focused on reuse of content, but recently attention has focused on reusable software tools and reusable activity structures. The former has led to the proposal of service-oriented architectures, and the latter has seen the development of the Learning Design specification. The authors suggest that there is a mutual dependency between the success of these two approaches, as complex Learning Designs require the ability to call on a range of tools, while remaining technology neutral.

The paper describes a project at the UK Open University, SLeD, which sought to develop a Learning Design player that would utilise the service-oriented approach. This acted both as a means of exploring some of the issues implicit within both approaches and also provided a practical tool. The SLeD system was successfully implemented in a different university, Liverpool Hope, demonstrating some of the principles of reuse.

Keywords

Learning Design, service-oriented architecture, virtual learning environments, e-learning, learning management systems

Aspects of reuse

Over recent years there has been a considerable push towards reuse and interoperability, both within the educational sector and in terms of broader data exchange. This desire for interoperability has several motivations underpinning it. Perhaps primary amongst these are cost considerations. As it became evident that e-learning was not a cheap alternative to face to face teaching, then the desire to reuse content grew (Weller 2004). The initial focus of reuse was on content, with the notion of learning objects, and growing an 'educational object economy'. Much of the initial efforts were focused on facilitating this reuse of content, with the resulting standards providing means of describing resources (metadata) and structures of resources (content packaging).

As well as reusing content, it makes financial sense to reuse software components in the development of larger systems. A related motivation is the convenience afforded by reusing existing components that have already been developed and tested, instead of creating each one from scratch. The rise and acceptance of open source software developments has also suggested a third motivation, namely that of quality. By allowing components to be reused in different contexts they are improved or adapted by a community of users, to become increasingly robust. This desire to reuse components has been reflected in the development of standards that seek to address this, for instance, the Tools Interoperability Specification by IMS. However, while there are generic web services standards, software interoperability is still an immature area with few robust, and reliable standards that can be used to guide particular implementations.

A possible way forward is reflected in a number of large projects focusing on the development of reusable, interoperable components, particularly in the Virtual Learning Environment (VLE) sector. Here, the notion of a technical architecture based around generic descriptions of services has been advanced, known as a Service-Oriented Architecture (SOA). Such architecture would facilitate the development of component VLEs comprised of a number of best of breed components that can plug together, instead of the more integrated, monolithic systems offered commercially. The viability of such component VLEs has been advanced by recent developments that seek to specify a generic, standards-based approach to VLEs, often focused around open source systems. These include the SAKAI initiative in the US and the JISC service-oriented architecture in the UK. The SAKAI project (<http://www.sakaiproject.org>), aims to deliver the following all as open-source:

The products of this project will include an Enterprise Services-based Portal, a complete Course Management System with sophisticated assessment tools, a Research Support Collaboration System, a Workflow Engine, and a Technology Portability Profile as a clear standard for writing future tools that can extend this core set of educational applications.

The JISC framework (Wilson et al., 2004) outlines the benefits and approach for adopting a service-oriented architecture, which can be seen as a means of viewing the integration of systems:

When we embark on this kind of analysis, identifying the parts of the MLE (Managed Learning Environment) at a more granular level than monolithic systems, then we eventually end up with a framework of service descriptions. We are no longer interested so much in replicating data between large systems, but instead focus on what kinds of services are needed in the overall architecture to provide certain kinds of behaviour from applications.

The last area of reuse is that of pedagogy, whereby activity structures, or Learning Designs, can be reused in different subject areas. The concept behind reusable Learning Designs is that an activity once specified clearly enough is reusable in a different subject matter, merely by changing the resources, for example, an online debate in political history has the same underlying structure as one in evolutionary psychology — it is only the resources that will alter. Reuse of Learning Designs is an attractive idea and has led to work on approaches to design for learning, activity templates and learning patterns to help understand how to describe activities. These activity descriptions will have greater value for reuse if they can be transferred to new contexts with changed content and changed sets of tools. Thus, pedagogic reuse provides an important motivation for both content reuse and tool reuse. In our view, this motivation is vital for acceptance of reuse into the teaching process and it is important to provide good representations of the resulting activities.

One approach to representing pedagogic design is IMS Learning Design (LD). This was developed from the Educational Modelling Language (EML) (Hummel et al., 2004) to provide an agreed specification for an approach to designing online activities and a computer readable XML representation of the resulting unit of learning. The representation links together the materials with roles for participants and sets of properties and conditions that enable people to work through the unit.

One of the strengths of LD is the use of a flexible format for storing the designs, however the concept of separating out the function of design from the delivery has proved powerful in itself. This has led to other ways to approach the design phase, for example Learning Activity Management System (LAMS) (Dalziel, 2003). In LAMS the designer can call on a custom set of tools, but this limits the designs to run in only the one environment. LD on the other hand has the potential to make use of whichever tools are available; however, that is only useful if a match can be made to the correct tool with the right capabilities to support the designed activity. To release this potential there needs to be an integration of the reuse of pedagogic designs with the supporting toolsets.

Learning Design and Service-Oriented Architectures

If the reuse of Learning Designs is to be realised, then it is likely to be because they meet three main motivations for reuse: they provide savings; can be more convenient than creating from scratch; and offer quality benefits. Such Learning Designs are likely to be reasonably complex and pedagogically rich, since relatively simple ones can be easily created, thus reducing the benefits of reuse. This complexity of structure will often lead to a requirement for the use of a range of tools and services. Currently only email, conference and search are specified in the LD guidelines. In order for complex designs to be created, a greater range of services needs to be described, along with the provision for adding to these.

If Learning Designs are to be reusable however, they need to remain neutral in terms of requiring specific tools, since they are not reusable if they require a specific tool. A service-oriented approach to services and tools is then especially relevant to a Learning Design perspective, and the two can be seen as interlinked. Environments configured as services then have the potential to accommodate LDs that call on specific instances of those services.

To realise this, three factors need to be in place:

- Generic descriptions of services that a Learning Design can interpret in order to create complex pathways through material. For example, all bulletin boards perform the same sorts of functions. By describing these, a design that utilises a bulletin board for an online debate with different roles (e.g. proponent, opposer, scribe) can be realised.
- A methodology for describing these services so that new ones can be added. This needs to encompass the means by which services are described, how LD recognises these and how the description or consensus about a description is arrived at.
- Tools, services and environments that are amenable to such an approach. This will include being able to expose the main functions of a tool to a Learning Design system without complex recoding of the initial software.

However, creating a generic service driven architecture is not easy. Despite much of the discussion surrounding this approach, there are few successful implementations. The Tasmanian LeAP project is a rare example (LeAP 2004) that uses a service-oriented approach to create a flexible VLE:

The project has guiding principles of interoperability and the use of standards for data and infrastructure. The preferred application architecture model uses a “service based infrastructure” approach. The reality is that the diversity of products within the educational computing environment makes it impossible to adopt a single approach to application architecture. LeAP considers it good practice to use existing services and create new services as application development progresses.

This may simply be a reflection of the relative immaturity of this approach. SAKAI is a relatively new consortium and has given itself a tight timescale to deliver the first version of their vision.

However, the lack of large-scale robust systems deploying generic service descriptions may also point to more fundamental problems, and maybe it remains an attractive theoretical construct that is difficult to realise in practical terms.

The question facing those working in this field then is to what extent a generic service description approach is achievable, and practical? If it is realisable then there are important implications for elearning, since it allows best of breed environments to be developed. As Wilbert Kraan of the UK's centre for educational technology interoperability standards (Kraan, 2004) comments:

It is becoming clear that common e-learning activities ... can't really be done by one application that has little or no knowledge of everything else on the network or the wider internet. It's also becoming clearer that a single system that tries to combine all such functions is unlikely to do all of them equally well. Furthermore, one size systems do not necessarily fit all institutions.

The opposing view is that while generic service descriptions are appealing from a theoretical and architectural perspective, they are impractical and inefficient. For example, in developing the LAMS tool, Dalziel (2005) suggests that in order to create tools that are meaningful from a Learning Design perspective — 'Learning Design aware' tools as he terms them — it was more practical to build the tools from scratch than re-engineer existing ones. LAMS provides the educational author with a number of tools, such as voting, discussion, quizzes, etc. Each of these components was built specifically for the LAMS editor, so that the sequencing of activities that LAMS sets out can be realised. Dalziel makes the distinction between 'rich' and 'minimal' component integration, arguing that for the necessary control and flow through a Learning Design driven environment, rich integration is the better option:

Richly integrated components, as demonstrated in LAMS, are technically more challenging to achieve initially, but provides a seamless, integrated environment for both teachers and learners, with better potential for reliable quality of service.

A generic description will always offer less functionality than a complete, bespoke service description and so much of the richness of individual programs is lost. In addition, the brokering of services required in such an approach leads to inefficient data handling and needless additional steps in the transaction path.

What is required therefore is both a means of testing some of the assumptions in the service oriented approach, and also a tool that can interpret the kind of generic service descriptions set in such an approach so that they can be used with LD.

Implementing Learning Design within a Service-Oriented Environment

LD is a well-defined XML format and the CopperCore system developed at the Open University Netherlands (OUNL) validates Learning Design packages, to check if they conform to the specification, and if not, indicates the problem areas. Thus, a Learning Design package could be passed through CopperCore, validated, and then passed on to another system in the knowledge that it met the requirements of the specification and was a proper package. This formed the basis for a project to develop an extended player system to present the LD to the user and call on appropriate services. The SLeD (originally Service-based Learning Design) project (<http://sled.open.ac.uk>) was a collaboration between the UK Open University (UKOU) and OUNL, funded by JISC as part of the Elearning Framework (ELF) programme, a joint initiative by the JISC in the UK and the Australian DEST (<http://www.elframework.org/>). For both the UKOU and the OUNL, Learning Design has three possible benefits:

- As a means of describing course design in a format that can be shared between academics and technical staff.

- As an audit trail of the design decisions, which can be reviewed as part of any quality assurance process.
- As a means of providing structure and support to students and tutors via the delivery mechanism. This is particularly acute when students are studying at a distance and attempting complicated activities.

All of these potential benefits require good tools and an understanding of how Learning Design can best be used. The SLeD project can thus be seen as an attempt to address at least part of this growing institutional interest in Learning Design (McAndrew & Weller, 2005). The output of the project was a software system that could 'play' Learning Designs, by interpreting them and making the appropriate calls to the various tools required by the particular Learning Design it was running. For instance, one design might simply require the sequential display of information to the user, which would all be handled through the SLeD system, while another design might require that users post messages in a forum system, which would require the player to call on whatever forum system the institution used.

Initial work was successful in integrating two types of forums into the SLeD player, and similarly two separate instantiations of a search function. The two forums were OpenText's FirstClass system, and the in-house Knowledge Network at the OUUK. The two search tools were Google, and the Knowledge Network again. The integration of these services demonstrated that both commercial and open systems could be called from the player, giving users a wide choice as to the actual implementation of any service they prefer.

A second project was initiated in April 2005, with further funding from JISC. The aim was to build on the success of the first project, particularly in extending and formalising the approach for integrating services while maintaining the architectural integrity of the system, and also to allow the integration of assessment packages within Learning Designs, specified in the IMS QTI (Question and Test Interoperability) standard.

The project extended the initial SLeD work in two key areas. Firstly, although LD had generated a lot of interest and enthusiasm, it was now at the stage where it needed to be put into practical use. The integration of assessment services into the Learning Designs was seen as a crucial factor in this. By upgrading CopperCore to validate IMS LD packages containing QTI this significant advance could be realised through a recognised core Learning Design system.

The second key area was to address the area of current weakness in the LD specification, namely the paucity of services that it can reference. The work in the SLeD project began to address this by developing a generic method for calling search and conferencing services. The second phase of the project further developed this approach and formalised it, to create a toolkit for service integration. The proposed approach was to use the QTI work as a test bed to develop a generic technical methodology for integrating services.

The experience of implementing bulletin boards in the initial SLeD project, and the work in integrating QTI calls led the project team to devise the architecture shown in Figure 1.

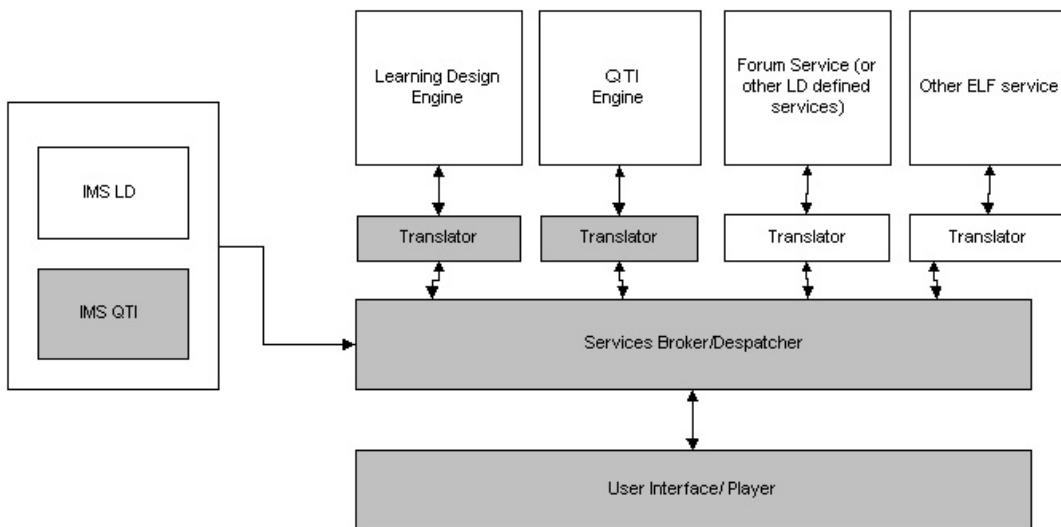


Figure 1: The SLeD architecture

The shading in the diagram represents the work of the second SLeD project, in extending the work undertaken in the initial project.

In this model the services broker houses what are termed ‘generic service descriptions’, that is, descriptions of any one service (for example, search) that apply across all the instantiations of that service. The LD engine (in our case CopperCore) is responsible for the validation of the package and the correct publication of it to the different services. The service dispatcher interprets the type of resource requested by the player and acts upon it. It contains the logic for synchronising the properties and calling the underlying services. In the case of the SLeD project the LD engine will be CopperCore, and the player is SLeD, but in this open architecture these could both be replaced with alternative systems. Similarly, the QTI engine should be any standards compliant engine. Other services might include forums (or bulletin boards), eportfolios, search, email, etc.

The player handles the display, coordination and user interface of the services to the user. Although generic service descriptions are stored in the broker, unless each individual service is set up to pass information back in exactly the required format, the generic descriptions will not be able to correctly handle the data. It is therefore necessary to have a small piece of application specific code that in effect acts as a translator between the application in question and the generic service description.

This model was successfully implemented and expanded to handle calls from eportfolio tools. However, in order to realise the sort of rich data handling required for a Learning Design approach, it is necessary not just to launch an application once, but to also pass data between that system and the player, so for instance it would need to know when a message has been posted or read in a forum. In order to achieve this it is necessary to develop an agreed language for describing these actions so that they can be specified within any LD unit of learning. Thus, for a new service such as eportfolios it requires a complete description of the type of functions such a service would perform, and then agreement in the community on this description.

SLeD in Use

The SLIDe project (<http://www.hope.ac.uk/slides>) at Liverpool Hope University used the SLeD system on a second-year assessed course in computing. The aims of the SLIDe project were to integrate the SLeD and CopperCore systems with existing university systems, and to evaluate both the technical performance and user experience of SLeD.

The project team created their own Learning Designs (using the Reload IMS Learning Design editor), which were then run using the SLeD system. In the creation of the Learning Designs some of the stages suggested in Section 3.2.1 of the IMS Best Practice and Implementation Guide (IMS Global Learning Consortium, 2003) were used (analysis, design and development). Figure 2 shows a sample fragment of the Learning Design for the course. This shows the student, who has been assigned to the role ‘learner’, working through the sequence of activities. The design also shows how the student interacts with the tutor, who, in this case, reviews progress at key points and allows progression through the SLeD system.

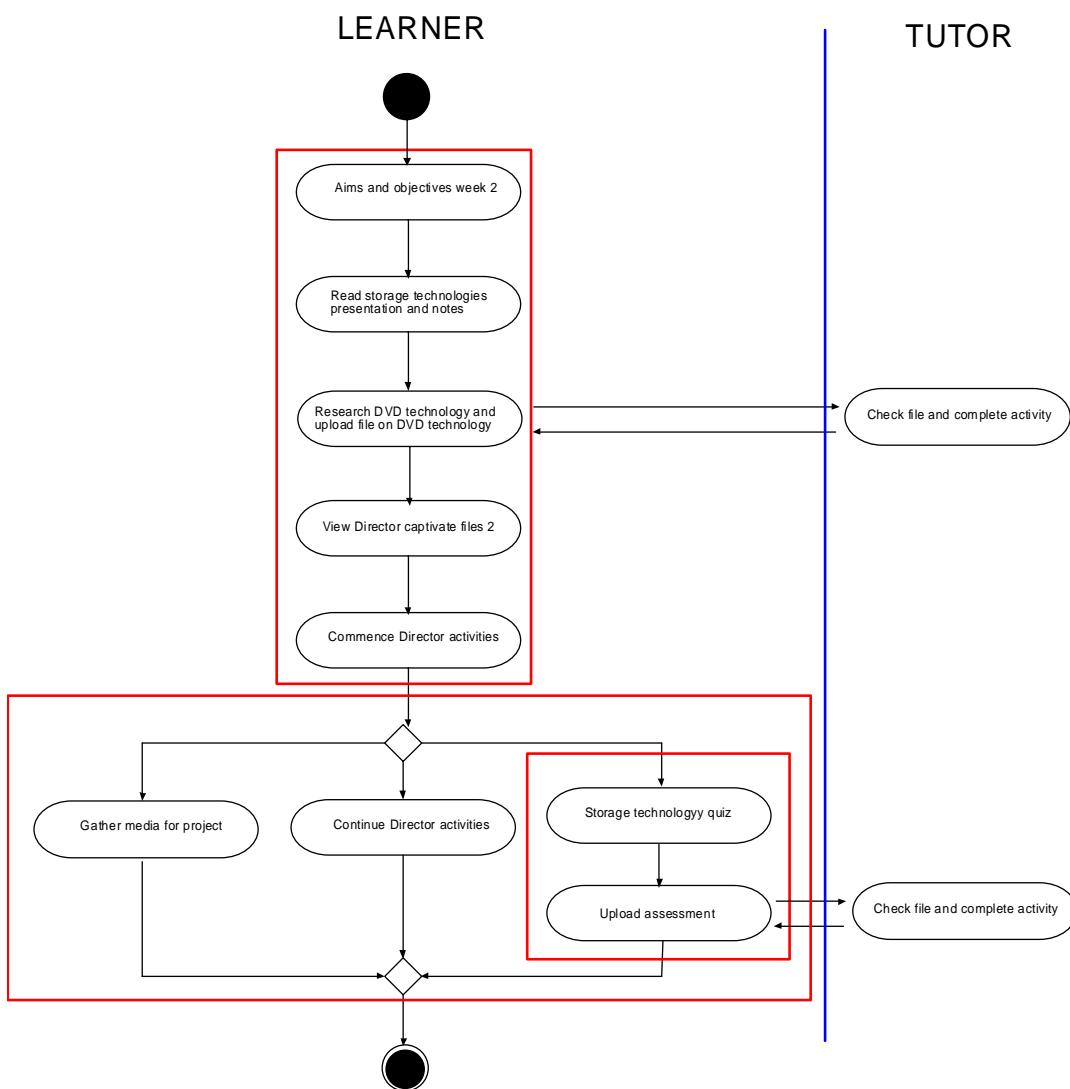


Figure 2: Sample activity diagram

The SLeD system was made available to learners through their normal institutional authentication. The trial was run during the first semester of academic year 2005/06 with a group of 15 students, studying a second-year module in Multimedia Design.

In terms of the technical aspects there were a number of small issues that were resolved in collaboration with the SLeD team. The main technical issue was that of performance, as the system could be slow to respond. Some of these problems were overcome, but this performance issue is a potential disadvantage of the service-oriented approach that SLeD embodies.

In terms of user evaluation, the fifteen students were asked to complete a questionnaire mid-way through the course and at course completion. There was no significant difference between the responses at these two stages. Generally, students were positive regarding the educational aid the system offered, but frustrated by reliability and performance issues. For example, using a Likert scale of 1 (minimum) to 5 (maximum) the statement 'I find it useful to be guided through the activities' had a mean score of 3.93, and the statement 'The software has helped me to learn' returned a mean score of 3.21, whereas 'The software is reliable' only had a mean score of 2.21.

From the open-ended questions, a typical positive response was 'SLeD for me is an excellent way of studying as you can progress through it at your own pace and you also have the advantage of going back over what you have done if you feel you need to ...'. A typical negative response was 'SLeD for me has been very frustrating. The http: error status screen sometimes restricts what I can actually do with SLeD'.

Overall, the project demonstrated that a Learning Design system can be used effectively to enhance learning with 'real' students, but that performance issues can detract from any benefits accrued.

Conclusions

In this paper we have looked at the growing interest in reuse in three main areas:

- Content — seen through the development of learning objects and repositories
- Tools — evidenced through service-oriented architecture approaches and projects such as SAKAI and JISC's elearning framework
- Pedagogy — realised through the IMS Learning Design specification.

It was argued that in order for Learning Design to be successful in its aim of reusing activity structures, the development of generic service descriptions was necessary. The argument could also be made from the perspective of the development of service-oriented architectures — such projects require a good deal of reengineering and development work, and thus they need to have considerable benefits in order to be worthwhile. One strong benefit is that they facilitate the sort of rich elearning experience based around an activity approach that Learning Design encourages, over the more content, instructivist approach afforded by many existing VLEs. Thus, there is a mutual dependency between the success of the two approaches.

The SLeD project sought to examine some of the issues inherent in this approach. It successfully developed a player and an architecture that could accommodate the Learning Design and SOA methodologies, which was deployed in another institution. However, although this project has offered a potential model for incorporating the generic service approach into the current environment, there remain a number of unanswered questions. Firstly, we have not yet determined the efficiency of such a system and whether there is a significant load in transfer of data. With many users it could be that such a system becomes too inefficient and significant time delays occur. Secondly, and perhaps more significantly, we have not explored the limitations of a generic service approach. While it is possible to derive a list of generic functions from a range of tools providing the same service, by necessity this ignores differences between them.

Thus, any particular richness, or subtle nuances of a specific program will be lost through this approach since only the generic services are required. This may be the price that is paid for any reusability. A specific instance can always provide a richer example than one that is created to be reused in multiple contexts. The issue is whether that additional richness is worth the cost.

Thirdly, the complexity of creating generic service descriptions should not be underestimated. In creating one for the eportfolio integration, it became apparent that in order for the system to do more than simply act as a ‘dumb’ interface to an eportfolio tool it would be necessary to devise a list of all the generic functions of eportfolio tools, describe these in programming terms and then get agreement from the eportfolio community and developers. Thus, although the endpoint may be desirable, the effort required to reach it may prove to be prohibitive.

Further work is required to extend the range of services that can be included, and thus to provide a more robust test of the methodology. In addition, there are issues that this project did not address, for instance the user interface of different systems, and the extent to which the player, or the initiating service controls this. While it may be possible for the player to offer a uniform user interface, by simply taking data in a web services format, this underestimates the significance of how the interface affects the user’s behaviour in the originating service. It may be that a software system with a different interface simply does not make sense to the user, or more likely, that it subtly alters their interaction with the system, so that users of the original program, and users of the Learning Design player version behave differently. There is still comparatively little work on the affordances of software and the subtle influences this can have on how a user interacts with a system, but in the general shift towards service-oriented architectures with their emphasis on underlying system functionality, it is important we do not overlook the nuances of interface design.

References

Dalziel, J. (2003). *Implementing learning design: The learning activity management system (LAMS)*. Paper presented at ASCILITE Conference 2003, Adelaide. Retrieved January 22, 2004, from <http://www.lamsinternational.com/documents/ASCILITE2003.Dalziel.Final.pdf>

Dalziel, J. (2005). *From reusable e-learning content to reusable learning designs: Lessons from LAMS*. Paper presented at the EDUCAUSE AustralAsia Conference, Auckland. Retrieved October 25, 2005, from <http://www.lamsfoundation.org/CD/html/resources/whitepapers/Dalziel.LAMS.doc>.

Hummel, H. G. K., Manderveld, J. M., Tattersall, C., & Koper, E. J. R. (2004). Educational modelling language and learning design: New opportunities for instructional reusability and personalised learning. *International Journal Learning Technology*, 1(1), 111–126.

IMS Global Learning Consortium (2003). *IMS Learning Design Best Practice and Implementation Guide*. Retrieved (n.d.), from http://www.imsglobal.org/learningdesign/LearningDesignv1p0/imsLearningDesign_bestv1p0.html

Kraan, W. (2004, October 8). *CETIS Quarterly Newsletter*.

LeAP Project Case Study. (2004, July 20). *Implementing web services in an education environment*. Retrieved (n.d.), from <http://www.education.tas.gov.au/admin/ict/projects/imsdoecasesstudy/LeAPPProjectCase.pdf>

McAndrew, P., & Weller, M. (2005). Applying learning design to supported open learning. In R. Koper & C. Tattersall (Eds.), *Learning design: A handbook on modelling and delivering networked education and training* (pp. 281–290). New York: Springer.

Weller, M. J. (2004). Learning objects and the e-learning cost dilemma. *Open Learning*, 19(3), 293–302.

Wilson, S., Olivier, B., Jeyes, S., Powell, A., & Franklin, T. (2004). *A technical framework to support e-learning*. Bristol, UK: JISC. Retrieved September 30, 2005, from http://www.jisc.ac.uk/uploaded_documents/Technical%20Framework%20feb04.doc

Acknowledgments

The SLeD development, SLeD demonstrator, SLeD-2 toolkit and SLIDe projects are funded by the UK JISC under the e-Learning Framework program. The authors would like to acknowledge the invaluable work of the partners in the SLeD project at the UKOU (Alex Little and Will Woods) and at the OUNL, in particular Rob Koper, Rob Nadolski, Hubert Vogten and Harrie Martens.

© Copyright Patrick McAndrew, Martin Weller and Mark Barrett-Baxendale [2006].

Please cite as: McAndrew, P., Weller, M., Barrett-Baxendale, M. (2006). Learning design and service oriented architectures: A mutual dependency? *Journal of Learning Design*, 1(3), 51-60.
<http://www.jld.qut.edu.au/>